

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ  
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНО-ДИСТАНЦІЙНОГО  
НАВЧАННЯ**

**ФОРМА НАВЧАННЯ ЗАОЧНА  
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ  
ІНФОРМАТИКИ**

**Допускається до захисту**

Завідувач кафедри \_\_\_\_\_ О.О. Ємець  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему**

**СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ  
«ФОРМАЛЬНІ ГРАМАТИКИ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО  
КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ»**

**зі спеціальності 122 «Комп'ютерні науки»**

**Виконавець роботи** Остапов Ігор Сергійович

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**Науковий керівник** к.ф.-м.н., доц. Черненко Оксана Олексіївна

\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 2021р.  
(підпис)

**ПОЛТАВА 2021 р.**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ .....	3
ВСТУП .....	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	9
2.1. Дистанційне навчання .....	9
2.2. Актуальність теми.....	13
3. ТЕОРЕТИЧНА ЧАСТИНА .....	15
3.1. Огляд матеріалу за темою роботи .....	15
3.2. Алгоритмізація задачі за темою роботи .....	21
3.3. Обґрунтування вибору програмних засобів для реалізації завдання роботи.....	32
4. ПРАКТИЧНА ЧАСТИНА .....	34
4.1. Розробка блок-схеми, яка підлягає програмуванню .....	34
4.2. Опис процесу програмної реалізації .....	35
4.3. Опис роботи тренажеру.....	38
ВИСНОВКИ.....	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	45
ДОДАТОК А.....	47

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
$G$	формальна породжувальна граматика
$V$	скінченна непорожня множина, яку називають алфавітом
$T$	підмножина $V$ , елементи якої називають термінальними (основними) символами
$S$	початковий символ ( $S \in V$ )
$P$	скінченна множина продукцій (або правил перетворення)
Netlify	універсальна платформа для автоматизації сучасних веб-проектів
Typescript	мова програмування

## ВСТУП

Сучасному суспільству потрібна масова якісна освіта, яка спроможна забезпечити зросли вимоги до споживача та виробника матеріальних і духовних благ. Виконати соціальне замовлення суспільства через збільшення асигнувань на освіту, збільшення кількості навчальних закладів та іншими традиційними способами не в змозі навіть заможні країни. Тому поява дистанційної освіти не випадкова, це закономірний етап розвитку та адаптації освіти до сучасних умов.

В Україні близько 30% навчальних закладів заявили про те, що вже мають або планують організувати навчання в режимі дистанційної освіти (ДО). Однак найчастіше за цим стоїть звичайна заочна форма навчання. Чим же відрізняється дистанційна освіта від інших видів отримання знань і професії? Насамперед, ДО – це відкрита система навчання, що передбачає активне спілкування між викладачем і студентом за допомогою сучасних технологій та мультимедіа. Така форма навчання дає свободу вибору місця, часу та темпу навчання.

Система ДО має ряд переваг і значно розширює коло потенційних студентів. Дистанційна форма навчання підходить майже всім, тому що дає можливість гармонійно поєднувати навчання та повсякденне життя.

Метою роботи є створення програмного забезпечення тренажеру з теми «Формальні граматики» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес дистанційного навчання дисциплінам.

Предметом розробки є тренажер з теми «Формальні граматики».

Перелік використаних методів - застосування математичних основ теорії програмування з теми «Формальні граматики», універсальна платформа для автоматизації сучасних веб-проектів Netlify, мова програмування Typescript, фреймворк Angular.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі. У другому розділі описано огляд дистанційного навчання, актуальність теми. В третьому розділі представлено огляд матеріалу за темою роботи, алгоритмізацію задачі за темою роботи, обґрунтування вибору програмних засобів. В четвертому розділі описано розробку блок-схеми, процес програмної реалізації, роботу тренажеру.

Обсяг пояснювальної записки: 51 стор., в т.ч. основна частина - 42 стор., джерела - 11 назв.

## 1. ПОСТАНОВКА ЗАДАЧІ

Основним завданням роботи є розробка тренажеру з теми «Формальні граматики» дистанційного навчального курсу «Теорія програмування».

Розглянемо основні завдання роботи:

- описати постановку задачі;
- описати інформаційний огляд;
- переглянути теоретичний матеріал з теми та навести основні його поняття для використання в тренажері;
- розробити алгоритм тренажеру;
- скласти блок-схему роботи алгоритму;
- описати мову програмування та інші технології, що використовувалися при розробці;
- описати процес реалізації основних етапів створення тренажеру;
- розробити тренажер.

Процес розробки складається з безлічі підпроцесів, деякі з яких показані нижче:

- аналіз вимог;
- специфікація програмного забезпечення;
- проєктування програми;
- програмування;
- тестування програми;
- впровадження тренажеру.

Оскільки тренажер буде використовуватися іноземними студентами в рамках вивчення дистанційного курсу «Теорія програмування», то потрібно розробити можливість запуску тренажеру відразу на іншій мові або її зміни під час виконання.

Завдання для розв'язку:

- Речення української мови можна вивести в граматиці  $G$ , де  
 $V = \{ \langle \text{речення} \rangle, \langle \text{група підмета} \rangle, \langle \text{група присудка} \rangle, \langle \text{означення} \rangle, \langle \text{підмет} \rangle, \langle \text{присудок} \rangle, \langle \text{додаток} \rangle, \text{молодий}, \text{нападник}, \text{забив}, \text{гол} \}$ ,  
 $T = \{ \text{молодий}, \text{нападник}, \text{забив}, \text{гол} \}$ ,  
 $S = \langle \text{речення} \rangle$ ,  
 $P = \{ \langle \text{речення} \rangle \rightarrow \langle \text{група підмета} \rangle \langle \text{група присудка} \rangle, \langle \text{група підмета} \rangle \rightarrow \langle \text{означення} \rangle \langle \text{підмет} \rangle, \langle \text{група присудка} \rangle \rightarrow \langle \text{присудок} \rangle \langle \text{додаток} \rangle, \langle \text{означення} \rangle \rightarrow \text{молодий}, \langle \text{підмет} \rangle \rightarrow \text{нападник}, \langle \text{присудок} \rangle \rightarrow \text{забив}, \langle \text{додаток} \rangle \rightarrow \text{гол} \}$ .

Ця схема породжує тільки одне речення, вкажіть його.

- Нехай  $G$  – граMATика з алфавітом  $V = \{S, A, a, b\}$ , множиною терміналів  $T = \{a, b\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow aA, S \rightarrow b, A \rightarrow aa\}$ . Знайти мову  $L(G)$ , породжувану цією граMATикою.

- Нехай  $G$  – граMATика з алфавітом  $V = \{S, 0, 1\}$ , множиною терміналів  $T = \{0, 1\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow 11S, S \rightarrow 0\}$ . Знайти  $L(G)$ .

- Щоб граMATика  $G$  з попереднього завдання породжувала нескінченну мову, у множині продукцій має бути принаймні одне рекурсивне правило. Вкажіть це правило.

- Знайти граматику, яка породжує мову  $\{0^m 1^m \mid m = 0, 1, 2, \dots\}$ .

- Наведемо дві такі граматики:

$$G_1: V = \{S, 0, 1\}, T = \{0, 1\}, P = \{S \rightarrow 0S, S \rightarrow S1, S \rightarrow \varepsilon\};$$

$$G_2: V = \{S, A, 0, 1\}, T = \{0, 1\}, P = \{S \rightarrow 0S, S \rightarrow 1A, S \rightarrow 1, A \rightarrow 1A, A \rightarrow 1, S \rightarrow \varepsilon\}.$$

Мова  $\{0^m 1^n \mid m, n = 0, 1, 2, \dots\}$  регулярна. Вкажіть граматiku, яка її породжує.

Мова  $\{0^m 1^m \mid m = 0, 1, 2, \dots\}$  контекстно вільна. оскільки вона породжена граматикою з продукціями. Вкажіть ці продукції.



## **2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

### **2.1. Дистанційне навчання**

Нововведення або інновації характерні для будь-якої професійної діяльності людини і тому природно стають предметом вивчення, аналізу та впровадження. Інновації самі по собі не виникають, вони є результатом наукових пошуків, передового педагогічного досвіду окремих викладачів і цілих колективів. Одним з видів інновацій в організації професійної освіти є введення дистанційного навчання [2].

Дистанційне навчання – нова організація освітнього процесу, що ґрунтується на використанні як кращих традиційних методів навчання, так і нових інформаційних та телекомунікаційних технологій, а також на принципах самостійного навчання, призначена для широких верств населення незалежно від матеріального забезпечення, місця проживання, стану здоров'я.

Дистанційне навчання дає змогу впроваджувати інтерактивні технології викладення матеріалу, здобувати повноцінну освіту, підвищувати кваліфікацію співробітників у територіально розподілених місцях. Процес навчання може відбуватися будь-де і будь-коли, єдина умова – доступ до мережі Інтернет.

На сьогоднішній день інформаційні технології є невід'ємною частиною нашого життя. Їх основу становить програмне забезпечення, яке проникло в усі сфери людського життя.

На сучасному етапі у зв'язку з появою нових інформаційних технологій спостерігаються наступні зміни у системі освіти :

1. формальний підхід до навчання замінюється когнитивним;
2. один основний підручник з курсу замінюється можливістю вибрати з великого розмаїття джерел інформації;
3. поява гіпертексту;

4. використання різноманітних технологій і саметих, що принесуть найбільшу вигоду студентам;
5. навчаючі технології не повинні одна одну заміняти, вони мають доповнювати одна одну.

Нині ми говоримо саме про використання вільного програмного забезпечення для дистанційного навчання у вищих навчальних закладах (ВНЗ) Адже останнім часом у багатьох країнах світу намітилася тенденція його використання. Цьому посприяло те, що рівень і якість стали набагато вищими чим кілька років назад.

Недостатнє фінансування освітніх установ змушує шукати рішення, що надають можливість заощадити під час купівлі обладнання, а вже купівля ліцензійного програмного забезпечення є достатньо великою проблемою. Тим більше зараз, коли є можливість забезпечити виконання існуючих вузівських програм виключно на базі вільно поширюваного програмного забезпечення. Найбільш очевидним способом розв'язання даної проблеми є використання безкоштовного і вільно поширюваного програмного забезпечення [3].

Термін вільне програмне забезпечення ввів Річард Столмен, засновник проекту GNU. Вільними називаються програми, автор яких опублікував їх у супроводі так званої «вільної ліцензії», що дозволяє:

1. користуватися програмою для будь-яких цілей і на необмеженій кількості комп'ютерів або місць в мережі;
2. безперешкодно отримувати доступ до її вихідних кодів;
3. виготовляти необмежену кількість додаткових її примірників;
4. модифікувати її як для власного користування, так і для поширення на тих же умовах [4].

Річард Столмен виділяє такі основні причини використання вільного програмного забезпечення у навчальному процесі:

1. вільне програмне забезпечення дозволить заощаджувати кошти. Так навіть у багатих країнах існують проблеми з фінансуванням

освіти. У бідних країнах використання вільного програмного забезпечення, яке можна вільно копіювати і поширювати, дозволить розвивати інформаційні технології;

2. вільне програмне забезпечення дає можливість студентам вивчати, як воно працює — вихідний код є доступним;

3. студенти матимуть змогу копіювати програмне забезпечення у вузах для домашнього використання, що дозволить уникати використання неліцензійного програмного забезпечення. [3]

На сьогоднішній день існує багато рішень для систем дистанційного навчання, відмінних технічними можливостями, наявністю і рівнем складності різних функціональних компонентів, наприклад, Oracle (i-Learning), IBM (Learning Space), WebCT, “Прометей” виробництва НІЦ АСКБ, e-Learning компанії “Гіперметод”, та засоби Open Source: MOODLE, ATutor, Dokeos, Claroline тощо. Проте ми б хотіли розглянути розповсюдження за принципом Open Source на умовах ліцензії GNU/GPL, систему MOODLE.

Автором ідеї і концепції системи MOODLE, а також її засновником є Мартін Даугіамас (Martin Dougiamas – доктор педагогічних наук з Curtin University Technology, Perth, Австралія). Головною його метою було створення системи, відмінної від доступних на ринку, а саме такої, яка б враховувала педагогічні аспекти, що базуються на основах пізнавальної психології, і, перш за все, однієї з її течій, що називається конструктивізмом.

Інтерфейс, допомога і документація системи MOODLE існують на декількох десятках мовах, які доступні і можуть бути вибрані і використані в разі потреби кожним користувачем.

Система повністю обслуговується з рівня стандартного веб-браузера. Не передбачає ніяких спеціальних вимог до устаткування і операційної системи. Є повністю безкоштовною. В поєднанні з відносно простою і добре описаною інсталяцією платформа MOODLE без спеціальних

труднощів може бути швидко встановлена на вузівському, шкільному або навіть власному сервері [5].

Серед основних можливостей використання системи, можна виділити наступні [5]:

1. Надання через мережу освітніх матеріалів всім бажаючим студентам або вибраній групі користувачів.
2. Забезпечення і підтримка можливості взаємного спілкування зарівно як між учнями/студентами, які беруть участь в курсі, так і між учнями/студентами і вчителем/ведучим.
3. Документування і збереження робіт, результатів дискусії, документування заданих питань і одержаних відповідей.
4. Надання інструментів, які забезпечують можливість здійснення поточного контролю і оцінки досягнень окремих учасників і доставку зворотних даних щодо кожної теми, у тому числі оцінок і відгуків (рецензій) на їх роботи.
5. Надання можливості аналізу участі і активності окремих учасників курсу, аналіз часу, присвяченого на роботу з матеріалами.

Отже, ми можемо побачити, що система MOODLE швидко розвивається і відіграє велику роль для дистанційного навчання у вищих начальних закладах. Вона відкриває студентам доступ до нетрадиційних джерел інформації, підвищує ефективність самостійної роботи, знаходження і закріплення різних професійних навичок, а викладачам дозволяє реалізовувати принципово нові форми і методи навчання із застосуванням концептуального і математичного моделювання явищ і процесів.

Система дистанційної освіти може і повинна зайняти своє місце в системі освіти, оскільки при грамотній її організації вона може забезпечити якісну освіту, що відповідає вимогам сучасного суспільства сьогодні.

## 2.2. Актуальність теми

Світ веб-розробки постійно змінюється. Досить важливим є уважно стежити за всім, що відбувається: актуальність існуючих фреймворків, вибір оптимального серед них, тенденції розробки та оформлення сайтів та інші.

Розробнику важливо бути в курсі поточних технологічних розробок, наскільки це можливо. Він повинен дивитися на можливості, але також враховувати можливі ризики, пов'язані з новими технологіями. Регулярні експерименти з різними структурами допомагають відчутти, як потенційні вигоди, так і обмеження.

Навчання у ВНЗ дає найголовніше вміння – самостійно знаходити та дослідити новий матеріал. Більшість студентів оцінюють курси як місце для отримання не знань, а поштовху для пошуку престижної роботи. Проте насправді більшість із них приведуть студента-слухача до конкретного проміжного фінішу, що завершиться сертифікатом. Найголовніша цінність будь-якого курсу – це не працевлаштування, а розвиток у студента критичного мислення та вміння думати не шаблонами, а опираючись на власні навички та досвід.

У зв'язку з високим попитом на онлайн-курси на просторах мережі Інтернет створюється велика кількість онлайн-тренажерів та посібників. Перевага даних ресурсів полягає в тому, що сам користувач вирішує, коли йому зручно витрачати час на виконання того чи іншого завдання. Онлайн-курси слідують такому принципу як дотримання часових рамок здачі завдання. У свою чергу, онлайн- тренажер/посібник не містять обмежень у часі.

На сьогоднішній день існує велика кількість тренажерів різного формату, серед яких наступні:

- тренажери для отримання навиків сліпого введення;
- тренажери для вивчення правил дорожнього руху;

- тренажери для вивчення іноземних мов;
- тренажери для вивчення мов програмування.

Тому майбутній онлайн-тренажер необхідно розробити в форматі, де користувач зможе обрати як тренажер для освоєння теоретичного матеріалу, так і тренажер із динамічними підказками або перейти до виконання практичного завдання для закріплення знань теоретичного матеріалу .

### 3. ТЕОРЕТИЧНА ЧАСТИНА

#### 3.1. Огляд матеріалу за темою роботи

Формальна граматика або просто граматика в теорії формальних мов — спосіб опису формальної мови, тобто виділення деякої підмножини з множини всіх слів деякого скінченного алфавіту. Розрізняють породжувальні і аналітичні граматики — перші ставлять правила, за допомогою яких можна побудувати будь-яке слово мови, а другі дозволяють по даному слову визначити, входить воно в мову чи ні. Формальні граматики були введені американським вченим, математиком та філософом, Н. Хомським у 50-тих роках XX сторіччя.

Алфавіт — скінченна множина символів.  $\varepsilon$  — порожній ланцюжок, слово, послідовність. Алфавітом є об'єднання алфавітів, перетин, різниця алфавітів. Нехай  $T$  — алфавіт, тоді:

- $T^+$  — множина усіх можливих послідовностей, що складені з елементів цього алфавіту крім порожньої послідовності  $\varepsilon$ .
- $T^*$  — множина усіх можливих послідовностей, що складені з елементів цього алфавіту, будь-якої довжини. Отже:
- $T^k$  — множина усіх можливих послідовностей, що складені з елементів цього алфавіту, довжини не більше  $k$ .

**Мова** в алфавіті  $T$  це множина ланцюжків скінченної довжини в цьому алфавіті. Зрозуміло, що кожна мова в алфавіті  $T$  є підмножиною множини  $T^*$ .

У лінгвістиці природних мов терміни „речення” та „слово” мають різний зміст; тому в математичній лінгвістиці послідовність символів зазвичай називають нейтральним терміном „ланцюжок”, а мову, яку розуміють як множину формальних ланцюжків, — формальною.

Означення 3.1. **Формальна породжувальна граматика**  $G$  (далі – **граматика**  $G$ ) – це формальна система, задана четвіркою об'єктів

$$G = (V, T, S, P),$$

де  $V$  – скінченна непорожня множина, яку називають **алфавітом** (або **словником**);

$T$  – її підмножина, елементи якої називають **термінальними** (**основними**) символами;

$S$  – **початковий символ** ( $S \in V$ );

$P$  – скінченна множина **продукцій** (або **правил перетворення**) вигляду  $\xi \rightarrow \eta$ , де  $\xi$  та  $\eta$  – ланцюжки над алфавітом  $V$ . Множину  $V \setminus T$  позначають  $N$ , її елементи називають **нетермінальними** (**допоміжними**) символами.

Формальні породжувальні граматики часто називають **граматиками з фразовою структурою**, **граматиками безпосередніх складових**. Термінальні символи часто називають **терміналами**, а нетермінальні – **нетерміналами**.

У теорії формальних граматик усталилися традиції позначень, яких ми будемо дотримуватись. Символи термінального алфавіту позначають малими латинськими буквами чи цифрами, символи нетермінального алфавіту – великими латинськими буквами, ланцюжки над алфавітом  $V$  – грецькими буквами. Довжину ланцюжка  $\alpha$  позначають  $l(\alpha)$  або  $|\alpha|$ . Нагадаємо, що множину всіх ланцюжків у алфавіті  $V$  позначають  $V^*$ .

Нас цікавитимуть ланцюжки, які можуть бути породжені продукціями граматики.

Означення 3.2. Нехай  $G = (V, T, S, P)$  – граматика, і нехай  $\alpha_0 = \sigma\xi\tau$ ,  $\xi \neq \varepsilon$  (тобто  $\alpha_0$  – конкатенація ланцюжків  $\sigma$ ,  $\xi$  та  $\tau$ ),  $\alpha_1 = \sigma\eta\tau$  – ланцюжки над  $V$ . Якщо  $\xi \rightarrow \eta$  – продукція граматики  $G$ , то говорять, що  $\alpha_1$  **безпосередньо виводиться** з  $\alpha_0$ , і записують  $\alpha_0 \Rightarrow \alpha_1$ .

Означення 3.3. Якщо  $\alpha_0, \alpha_1, \dots, \alpha_n$  – ланцюжки над алфавітом  $V$  такі, що  $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n$ , то говорять, що  $\alpha_n$  **виводиться**



з  $\alpha_0$ , і використовують запис  $\alpha_0 \Rightarrow \alpha_n$ . Послідовність кроків для отримання  $\alpha_n$  з  $\alpha_0$  називають **виведенням**.

Речення української мови з попереднього прикладу можна вивести в граматиці  $G$ , де

$V = \{ \langle \text{речення} \rangle, \langle \text{група підмета} \rangle, \langle \text{група присудка} \rangle, \langle \text{означення} \rangle, \langle \text{підмет} \rangle, \langle \text{присудок} \rangle, \langle \text{додаток} \rangle, \text{молодий, нападник, забив, гол} \},$

$T = \{ \text{молодий, нападник, забив, гол} \},$

$S = \langle \text{речення} \rangle,$

$P = \{ \langle \text{речення} \rangle \rightarrow \langle \text{група підмета} \rangle \langle \text{група присудка} \rangle,$

$\langle \text{група підмета} \rangle \rightarrow \langle \text{означення} \rangle \langle \text{підмет} \rangle,$

$\langle \text{група присудка} \rangle \rightarrow \langle \text{присудок} \rangle \langle \text{додаток} \rangle,$

$\langle \text{означення} \rangle \rightarrow \text{молодий},$

$\langle \text{підмет} \rangle \rightarrow \text{нападник},$

$\langle \text{присудок} \rangle \rightarrow \text{забив},$

$\langle \text{додаток} \rangle \rightarrow \text{гол} \}.$

Ця схема породжує тільки одне речення, тому її можна замінити на

$L = \{ \text{молодий нападник забив гол} \}.$

Якщо ми захочемо розширити мову, щоб увести до неї речення з означеннями «талановитий», «надійний», присудком «віддав», і додатками «пас», «м'яч» (тоді мова  $L$  матиме вісімнадцять речень), то це можна зробити відповідним розширенням алфавіту та додаванням лише п'яти продукцій до множини  $P$ .

Означення 3.4. Мовою, породжуваною граматиною  $G = (V, T, S, P)$ , називають множину всіх ланцюжків терміналів, які виводяться з початкового символу  $S$ . Її позначають  $L(G)$ . Отже,

$$L(G) = \{ \omega \in T^* / S \Rightarrow \omega \}$$

де  $T$  — множина всіх ланцюжків терміналів, включаючи порожній ланцюжок.

Нехай  $G$  – граматика з алфавітом  $V = \{S, A, a, b\}$ , множиною терміналів  $T = \{a, b\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow aA, S \rightarrow b, A \rightarrow aa\}$ . Знайдемо мову  $L(G)$ , породжувану цією граматикою.

Із початкового символу  $S$  можна вивести ланцюжок  $aA$  за допомогою продукції  $S \rightarrow aA$  чи застосувати продукцію  $S \rightarrow b$ , щоб вивести  $b$ . З  $aA$ , скориставшись продукцією  $A \rightarrow aa$ , можна вивести ланцюжок  $aaa$ . Ніяких інших ланцюжків вивести неможливо. Отже,  $L(G) = \{b, aaa\}$ .

Нехай  $G$  – граматика з алфавітом  $V = \{S, 0, 1\}$ , множиною терміналів  $T = \{0, 1\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow 11S, S \rightarrow 0\}$ . Знайдемо  $L(G)$ .

Із початкового символу  $S$  одержимо  $0$  (за допомогою другої продукції) чи  $11S$  (за допомогою першої). З  $11S$  можна отримати  $110$  або  $1111S$ , з  $1111S$  виводиться  $11110$  або  $111111S$ . Отже, після кожного виведення ми додаємо дві одиниці в кінець ланцюжка чи закінчуємо ланцюжок нулем, тобто  $L(G) = \{0, 110, 11110, 1111110, \dots\}$  – це множина всіх ланцюжків із парною кількістю тільки  $1$ , після яких (у кінці) обов'язково є один  $0$ .

Ми отримали нескінченну мову  $L(G)$ , яка складається з нескінченної кількості ланцюжків. Щоб граматика  $G$  породжувала нескінченну мову, у множині продукцій має бути принаймні одне рекурсивне правило (наприклад,  $S \rightarrow 11S$ ).

Дуже важлива проблема побудови граматики для заданої мови.

Наприклад, знайдемо граматiku, яка породжує мову  $\{0^m 1^m \mid m = 0, 1, 2, \dots\}$ . Потрібні дві продукції, щоб побудувати ланцюжок із якоїсь кількості  $0$ , після яких є така сама кількість  $1$ . Перша продукція збільшує ланцюжок зсередини, додаючи зліва  $0$ , а справа –  $1$ . Друга продукція замінює початковий символ  $S$  на порожній ланцюжок  $\varepsilon$ . Одержимо граматiku  $G = \{V, T, S, P\}$ , де  $V = \{0, 1, S\}$ ,  $T = \{0, 1\}$ ,  $S$  – початковий символ,  $P = \{S \rightarrow 0S1, S \rightarrow \varepsilon\}$ .

Знайдено породжувальну граматику для мови  $\{0^m 1^n \mid m, n = 0, 1, 2, \dots\}$ .

Наведемо дві такі граматики:

$$G_1: V = \{S, 0, 1\}, T = \{0, 1\}, P = \{S \rightarrow 0S, S \rightarrow S1, S \rightarrow \varepsilon\};$$

$$G_2: V = \{S, A, 0, 1\}, T = \{0, 1\}, P = \{S \rightarrow 0S, S \rightarrow 1A, S \rightarrow 1, A \rightarrow 1A, A \rightarrow 1, S \rightarrow \varepsilon\}.$$

Означення 3.5. Граматики  $G_1$  та  $G_2$  називають **еквівалентними**, якщо  $L(G_1) = L(G_2)$ .

Продукція граматики дає змогу замінити одну послідовність символів іншою. Граматики класифікують за типами продукцій. Розглянемо класифікацію, яку запропонував американський математик і лінгвіст Н. Хомскі (табл. 3.1).

Таблиця 3.1 – Класифікація за Хомскі

Тип граматики	Обмеження на продукції $\xi \rightarrow \eta$
0	немає обмежень
1	$ \xi  \leq  \eta $ чи $\eta = \varepsilon$
2	$\xi = A$ , де $A$ – нетермінальний символ
3	$\xi = A$ , причому $\eta = aB$ чи $\eta = a$ , де $A, B$ – нетермінальні символи, $a$ – термінальний символ, або $S \rightarrow \varepsilon$

Принцип цієї класифікації полягає в тому, що на продукції накладено певні обмеження. Із цих означень випливає, що кожна граматика типу  $n$ , де  $n = 1, 2, 3$ , являє собою граматику типу  $n - 1$ . Граматики типу 2 називають **контекстно вільними**, бо нетермінал  $A$  в лівій частині продукції  $A \rightarrow \eta$  можна замінити ланцюжком  $\eta$  в довільному оточенні щоразу, коли він зустрічається, тобто незалежно від контексту. Мову, яку породжує граматика типу 2, називають **контекстно вільною**.

Якщо в множині продукцій  $P$  є продукція вигляду  $\mu\delta \rightarrow \nu\delta$  (але не  $\mu \rightarrow \nu$ ),  $|\mu| \leq |\nu|$ , то граматику називають граматикою типу 1, або

**контекстно залежною**, оскільки  $\mu$  можна замінити на  $\nu$  лише в оточенні ланцюжків  $\gamma \dots \delta$ , тобто у відповідному контексті. Мову, яку породжує граматика типу 1, називають **контекстно залежною**.

Граматика типу 3 називають **регулярною**. У ній можуть бути лише продукції  $A \rightarrow aB$ ,  $A \rightarrow a$ ,  $S \rightarrow \epsilon$ , де  $A, B$  – нетермінальні символи,  $a$  – термінальний символ. Мову, яку породжує граматика типу 3, називають **регулярною**.

На рис. 3.1 наведено діаграму, яка показує співвідношення між граматиками різних типів.

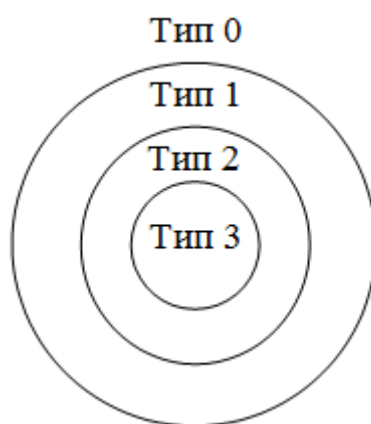


Рисунок 3.1 – Діаграма співвідношення між граматиками різних типів

Мова  $\{0^m 1^n \mid m, n = 0, 1, 2, \dots\}$  регулярна, бо її породжує регулярна граматика  $G_2$ .

Мова  $\{0^m 1^m \mid m = 0, 1, 2, \dots\}$  контекстно вільна, оскільки вона породжена граматикою з продукціями  $S \rightarrow 0S1$  та  $S \rightarrow \epsilon$ .

Приклад контекстно вільної мови – мова булевих формул зі змінними  $a, b, c$ , яку породжує контекстно вільна граматика  $G = (V, T, S, P)$ , де  $V = \{S, a, b, c, \vee, \wedge, \neg, (, )\}$ ,  $T = \{a, b, c, \vee, \wedge, \neg, (, )\}$ , а множина продукцій  $P = \{S \rightarrow (S \vee S), S \rightarrow (S \wedge S), S \rightarrow \neg S, S \rightarrow a, S \rightarrow b, S \rightarrow c\}$ .

### *Використання формальних граматик*

Формальні граматика - це дуже потужний математичний інструмент, який використовується в математичній та комп'ютерній лінгвістиці, описі мов програмування, розробці компіляторів в теорії програмування.

Найбільш вживаними є КВ-граматики і регулярні, бо їх найлегше описати алгоритмічно.

Сама по собі концепція формальних граматик доволі гнучка, тому не дивно, що з'явилося багато інших інструментів, що розширюють використання та потужність КВ-граматик і граматик третього типу. Наприклад, атрибутні граматики, LL-к граматики, скінченні автомати, регулярні вирази та множини. [6-8]

### 3.2. Алгоритмізація задачі за темою роботи

**Крок 0.** Користувачу виводиться назва тренажеру, пропонується розпочати проходження тестів.

**Крок 1.** Користувачу відображається умова: «Формальна граматика або просто граматика в теорії формальних мов — спосіб опису формальної мови, тобто виділення деякої підмножини з множини всіх слів деякого скінченного алфавіту. Розрізняють породжувальні і аналітичні граматики.

Виберіть відповідь» та наводяться варіанти відповіді:

- Перші ставлять правила, за допомогою яких можна побудувати будь-яке слово мови, а другі дозволяють по даному слову визначити, входить воно в мову чи ні;
- Перші дозволяють по даному слову визначити, входить воно в мову чи ні, а другі ставлять правила, за допомогою яких можна побудувати будь-яке слово мови.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – перший варіант».

**Крок 2.** Користувачу відображається умова: «Формальна породжувальна граматика  $G$  – це формальна система, задана \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- трійкою об'єктів  $G = (V, S, P)$ ;

- четвіркою об'єктів  $G = (V, T, S, P)$ ;
- п'ятіркою об'єктів  $G = (V, T, S, P, Y)$ .

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – другий варіант».

**Крок 3.** Користувачу відображається умова: «Формальна породжувальна граматика  $G$  (далі – граматика  $G$ ) – це формальна система, задана четвіркою об'єктів  $G = (V, T, S, P)$ .

Встановіть відповідність.» та наводяться варіанти:

- $V$ ;
  - $T$ ;
  - $S$ ;
  - $P$ .
1. початковий символ ( $S \in V$ );
  2. підмножина  $V$ , елементи якої називають термінальними (основними) символами;
  3. скінченна множина продукцій (або правил перетворення) вигляду  $\xi \rightarrow \eta$ , де  $\xi$  та  $\eta$  – ланцюжки над алфавітом  $V$ ;
  4. скінченна непорожня множина, яку називають алфавітом (або словником).

Якщо встановлено вірну відповідність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

$V$  – скінченна непорожня множина, яку називають алфавітом (або словником);

$T$  – її підмножина, елементи якої називають термінальними (основними) символами;

$S$  – початковий символ ( $S \in V$ );

$P$  – скінченна множина продукцій (або правил перетворення) вигляду  $\xi \rightarrow \eta$ , де  $\xi$  та  $\eta$  – ланцюжки над алфавітом  $V$ .».

**Крок 4.** Користувачу відображається умова: «Формальні породжувальні граматики часто називають \_\_\_\_\_».

Вставте продовження речення.» та наводяться варіанти відповіді:

- граматики з фразовою структурою;
- граматики безпосередніх складових;
- обидва варіанти правильні.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – третій варіант».

**Крок 5.** Користувачу відображається умова: «Термінальні символи часто називають \_\_\_\_\_. Нетермінальні символи часто називають \_\_\_\_\_».

Вставте пропущені слова» та наводяться варіанти:

- нетерміналами;
- просто символами;
- терміналами.

Якщо встановлено вірну відповідність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

Термінальні символи часто називають терміналами, а нетермінальні – нетерміналами».

**Крок 6.** Користувачу відображається умова: «У теорії формальних граматик усталилися традиції позначень, яких ми будемо дотримуватись. Символи термінального алфавіту позначають \_\_\_\_\_».

Вставте продовження речення.» та наводяться варіанти відповіді:

- грецькими буквами;
- великими латинськими буквами;
- малими латинськими буквами;
- цифрами.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – третій і четвертий варіанти».

**Крок 7.** Користувачу відображається умова: «У теорії формальних граматик усталилися традиції позначень, яких ми будемо дотримуватись. Символи нетермінального алфавіту позначають \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- грецькими буквами;
- великими латинськими буквами;
- малими латинськими буквами;
- цифрами.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – другий варіант».

**Крок 8.** Користувачу відображається умова: «У теорії формальних граматик усталилися традиції позначень, яких ми будемо дотримуватись. Ланцюжки над алфавітом  $V$  позначають \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- грецькими буквами;
- великими латинськими буквами;
- малими латинськими буквами;
- цифрами.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – перший варіант».

**Крок 9.** Користувачу відображається умова: «Довжину ланцюжка  $\alpha$  позначають \_\_\_\_\_.

Виберіть відповідь.» та наводяться варіанти відповіді:

- $l(\alpha)$ ;
- $\alpha$ ;



- $|\alpha|$ ;
- $(\alpha)$ .

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – перший і третій варіанти».

**Крок 10.** Користувачу відображається умова: «Нехай  $G = (V, T, S, P)$  – граматика, і нехай  $\alpha_0 = \sigma\xi\tau$ ,  $\xi \neq \varepsilon$  (тобто  $\alpha_0$  – конкатенація ланцюжків  $\sigma$ ,  $\xi$  та  $\tau$ ),  $\alpha_1 = \sigma\eta\tau$  – ланцюжки над  $V$ . Якщо  $\xi \rightarrow \eta$  – продукція граматики  $G$ , то говорять, що \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- $\alpha_0$  безпосередньо виводиться з  $\alpha_1$ , і записують  $\alpha_0 \Rightarrow \alpha_1$ ;
- $\alpha_1$  безпосередньо виводиться з  $\alpha_0$ , і записують  $\alpha_0 \Rightarrow \alpha_1$ ;
- $\alpha_1$  безпосередньо виводиться з  $\alpha_0$ , і записують  $\alpha_1 \Rightarrow \alpha_0$ .

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – другий варіант».

**Крок 11.** Користувачу відображається умова: «Якщо  $\alpha_0, \alpha_1, \dots, \alpha_n$  – ланцюжки над алфавітом  $V$  такі, що  $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_{n-1} \Rightarrow \alpha_n$ , то говорять, що \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- $\alpha_n$  виводиться з  $\alpha_0$ , і використовують запис  $\alpha_0 \Rightarrow \alpha_n$ ;
- $\alpha_n$  виводиться з  $\alpha_0$ , і використовують запис  $\alpha_n \Rightarrow \alpha_0$ ;
- $\alpha_0$  виводиться з  $\alpha_n$ , і використовують запис  $\alpha_0 \Rightarrow \alpha_n$ .

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – перший варіант».

**Крок 12.** Користувачу відображається умова: «Послідовність кроків для отримання  $\alpha_n$  з  $\alpha_0$  називають \_\_\_\_\_.

Вставте продовження речення.» та наводяться варіанти відповіді:

- введенням;
- виведенням;
- обидва варіанти правильні.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – другий варіант».

**Крок 13.** Користувачу відображається умова: «Речення української мови можна вивести в граматиці  $G$ , де

$V = \{ \langle \text{речення} \rangle, \langle \text{група підмета} \rangle, \langle \text{група присудка} \rangle, \langle \text{означення} \rangle, \langle \text{підмет} \rangle, \langle \text{присудок} \rangle, \langle \text{додаток} \rangle, \text{молодий, нападник, забив, гол} \},$

$T = \{ \text{молодий, нападник, забив, гол} \},$

$S = \langle \text{речення} \rangle,$

$P = \{ \langle \text{речення} \rangle \rightarrow \langle \text{група підмета} \rangle \langle \text{група присудка} \rangle,$

$\langle \text{група підмета} \rangle \rightarrow \langle \text{означення} \rangle \langle \text{підмет} \rangle,$

$\langle \text{група присудка} \rangle \rightarrow \langle \text{присудок} \rangle \langle \text{додаток} \rangle,$

$\langle \text{означення} \rangle \rightarrow \text{молодий},$

$\langle \text{підмет} \rangle \rightarrow \text{нападник},$

$\langle \text{присудок} \rangle \rightarrow \text{забив},$

$\langle \text{додаток} \rangle \rightarrow \text{гол} \}.$

Ця схема породжує речення, тому її можна замінити на  $L = \{ \text{_____} \}.$

Складіть речення» та наводяться варіанти відповіді:

- молодий;
- нападник;
- забив;
- гол.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –  $L = \{ \text{молодий нападник забив гол} \}.$ ».

**Крок 14.** Користувачу відображається означення: «Мовою, породжуваною граматикою  $G = (V, T, S, P)$ , називають множину всіх ланцюжків терміналів, які виводяться з початкового символу  $S$ . Її позначають  $L(G)$ . Отже,

$$L(G) = \left\{ \omega \in T^* / S \Rightarrow \omega \right\}$$

де  $T$  — множина всіх ланцюжків терміналів, включаючи порожній ланцюжок.».

Перехід на наступний крок.

**Крок 15.** Користувачу відображається умова: «Нехай  $G$  – граматика з алфавітом  $V = \{S, A, a, b\}$ , множиною терміналів  $T = \{a, b\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow aA, S \rightarrow b, A \rightarrow aa\}$ . Знайдемо мову  $L(G)$ , породжувану цією граматикою.

Встановіть послідовність кроків.» та наводяться варіанти:

- із початкового символу  $S$  можна вивести ланцюжок  $aA$  за допомогою продукції  $S \rightarrow aA$ ;
- з  $aA$ , скориставшись продукцією  $A \rightarrow aa$ , можна вивести ланцюжок  $aaa$ ;
- застосувати продукцію  $S \rightarrow b$ , щоб вивести  $b$ ;
- ніяких інших ланцюжків вивести неможливо. Отже,  $L(G) = \{b, aaa\}$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. застосувати продукцію  $S \rightarrow b$ , щоб вивести  $b$ ».

**Крок 16.** Користувачу вибирає наступний крок послідовності:

- із початкового символу  $S$  можна вивести ланцюжок  $aA$  за допомогою продукції  $S \rightarrow aA$ ;
- з  $aA$ , скориставшись продукцією  $A \rightarrow aa$ , можна вивести ланцюжок  $aaa$ ;

- ніяких інших ланцюжків вивести неможливо. Отже,  $L(G) = \{b, aaa\}$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. застосувати продукцію  $S \rightarrow b$ , щоб вивести  $b$ ;
2. із початкового символу  $S$  можна вивести ланцюжок  $aA$  за допомогою продукції  $S \rightarrow aA$ ».

**Крок 17.** Користувачу вибирає наступний крок послідовності:

- з  $aA$ , скориставшись продукцією  $A \rightarrow aa$ , можна вивести ланцюжок  $aaa$ ;
- ніяких інших ланцюжків вивести неможливо. Отже,  $L(G) = \{b, aaa\}$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. застосувати продукцію  $S \rightarrow b$ , щоб вивести  $b$ ;
2. із початкового символу  $S$  можна вивести ланцюжок  $aA$  за допомогою продукції  $S \rightarrow aA$ ;
3. з  $aA$ , скориставшись продукцією  $A \rightarrow aa$ , можна вивести ланцюжок  $aaa$ ;
4. ніяких інших ланцюжків вивести неможливо. Отже,  $L(G) = \{b, aaa\}$ ».

**Крок 18.** Користувачу вказує останній крок послідовності. Перехід на наступний крок.

**Крок 19.** Користувачу відображається умова: «Нехай  $G$  – граматика з алфавітом  $V = \{S, 0, 1\}$ , множиною терміналів  $T = \{0, 1\}$ , початковим символом  $S$  і множиною продукцій  $P = \{S \rightarrow 11S, S \rightarrow 0\}$ . Знайдемо  $L(G)$ .

Встановіть послідовність кроків.» та наводяться варіанти:

- з  $11S$  можна отримати  $110$  або  $1111S$ ;
- із початкового символу  $S$  одержимо  $0$  (за допомогою другої продукції) чи  $11S$  (за допомогою першої);

- після кожного виведення ми додаємо дві одиниці в кінець ланцюжка чи закінчуємо ланцюжок нулем, тобто  $L(G) = \{0, 110, 11110, 1111110, \dots\}$ ;
- з  $1111S$  виводиться  $11110$  або  $111111S$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. із початкового символу  $S$  одержимо  $0$  (за допомогою другої продукції) чи  $11S$  (за допомогою першої);».

**Крок 20.** Користувачу вибирає наступний крок послідовності:

- з  $11S$  можна отримати  $110$  або  $1111S$ ;
- після кожного виведення ми додаємо дві одиниці в кінець ланцюжка чи закінчуємо ланцюжок нулем, тобто  $L(G) = \{0, 110, 11110, 1111110, \dots\}$ ;
- з  $1111S$  виводиться  $11110$  або  $111111S$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. із початкового символу  $S$  одержимо  $0$  (за допомогою другої продукції) чи  $11S$  (за допомогою першої);
2. з  $11S$  можна отримати  $110$  або  $1111S$ ;».

**Крок 21.** Користувачу вибирає наступний крок послідовності:

- після кожного виведення ми додаємо дві одиниці в кінець ланцюжка чи закінчуємо ланцюжок нулем, тобто  $L(G) = \{0, 110, 11110, 1111110, \dots\}$ ;
- з  $1111S$  виводиться  $11110$  або  $111111S$ .

Якщо вказано вірну послідовність, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь –

1. із початкового символу  $S$  одержимо  $0$  (за допомогою другої продукції) чи  $11S$  (за допомогою першої);
2. з  $11S$  можна отримати  $110$  або  $1111S$ ;
3. з  $1111S$  виводиться  $11110$  або  $111111S$ ;

4. після кожного виведення ми додаємо дві одиниці в кінець ланцюжка чи закінчуємо ланцюжок нулем, тобто  $L(G) = \{0, 110, 11110, 1111110, \dots\}$ ».

**Крок 22.** Користувачу вказує останній крок послідовності. Перехід на наступний крок.

**Крок 23.** Користувачу відображається умова: «Продукція граматики дає змогу замінити одну послідовність символів іншою. Граматики класифікують за типами продукцій. Розглянемо класифікацію, яку запропонував американський математик і лінгвіст Н. Хомскі.

Встановіть відповідність.» та наводяться варіанти:

Тип граматики

- 0;
- 1;
- 2;
- 3.

Обмеження на продукції  $\xi \rightarrow \eta$

1.  $|\xi| \leq |\eta|$  чи  $\eta = \varepsilon$ ;
2. немає обмежень;
3.  $\xi = A$ , причому  $\eta = aB$  чи  $\eta = a$ , де  $A, B$  – нетермінальні символи,  $a$  – термінальний символ, або  $S \rightarrow \varepsilon$ ;
4.  $\xi = A$ , де  $A$  – нетермінальний символ).

Якщо встановлено вірну відповідність, то перехід на наступний крок.

Інакше – відображається помилка: «Помилка! Правильна відповідь –

Тип граматики	Обмеження на продукції $\xi \rightarrow \eta$
0	немає обмежень
1	$ \xi  \leq  \eta $ чи $\eta = \varepsilon$
2	$\xi = A$ , де $A$ – нетермінальний символ
3	$\xi = A$ , причому $\eta = aB$ чи $\eta = a$ , де $A, B$ – нетермінальні символи, $a$ – термінальний символ, або $S \rightarrow \varepsilon$

».

**Крок 24.** Користувачу відображається означення: «Формальні граматики - це дуже потужний математичний інструмент, який використовується в математичній та комп'ютерній лінгвістиці, описі мов програмування, розробці компіляторів в теорії програмування. Найбільш вживаними є КВ-граматики і регулярні, бо їх найлегше описати алгоритмічно».

Перехід на наступний крок.

**Крок 25.** Користувачу відображається умова: «Сама по собі концепція формальних граматик доволі гнучка, тому не дивно, що з'явилося багато інших інструментів, що розширюють використання та потужність КВ-граматик і граматик третього типу. Вкажіть приклади.

Виберіть відповідь.» та наводяться варіанти відповіді:

- атрибутні граматики;
- LL-k граматики;
- скінченні автомати;
- регулярні вирази;
- множини;
- всі варіанти правильні.

Якщо вказано вірну відповідь, то перехід на наступний крок. Інакше – відображається помилка: «Помилка! Правильна відповідь – останній варіант».

**Крок 26.** Користувачу виводиться повідомлення про завершення проходження тестів: «Вітаємо! Ви пройшли всі тести з теми «Формальні граматики», Ваша оцінка:», вказується один із варіантів в залежності від кількості допущених помилок:

- Відмінно;
- Дуже добре;
- Добре;
- Задовільно;
- Незадовільно.

### **3.3. Обґрунтування вибору програмних засобів для реалізації завдання роботи**

Netlify - це популярний сервіс хостингу і розгортання для проектів статичних сайтів. На відміну від інших хостингових платформ, Netlify працює практично з будь-яким генератором статичних сайтів, а не тільки з Jekyll або Sphinx.

Netlify надає CD для проекту. Можна зберігати контент в GitHub, GitLab або Bitbucket, зв'язати його з Netlify, і Netlify буде збирати заново сайт щоразу, при внесенні змін.

Netlify пропонує не тільки безкоштовний рівень з функціями, схожими на GitHub Pages, але також дозволяє переходити на рівень Pro, Business або Enterprise. З Netlify можна отримати попередній перегляд розгортання, відкати, обробку форм, розподілену мережу доставки контенту (CDN), нескінченну масштабованість, SSL, програмований API, CLI і багато іншого.

TypeScript — мова програмування, представлена Microsoft восени 2012; позиціонується як засіб розробки веб-застосунків, що розширює можливості JavaScript.

Розробником мови TypeScript є Андерс Гейлсберг (англ. Anders Hejlsberg), який створив раніше C#, Turbo Pascal і Delphi.

Код експериментального компілятора, котрий транслює код TypeScript в представлення JavaScript, поширюється під ліцензією Apache, розробка ведеться в публічному репозиторії через сервіс CodePlex. Специфікації мови відкриті і опубліковані в рамках угоди Open Web Foundation Specification Agreement (OWFa 1.0).

TypeScript є зворотно сумісним з JavaScript. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому



сучасному браузері або використовувати спільно з серверною платформою Node.js [9].

Переваги над JavaScript:

- можливість явного визначення типів (статична типізація),
- підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
- підтримка підключення модулів.

Angular представляє фреймворк від компанії Google для створення клієнтських додатків. Перш за все він націлений на розробку SPA-рішень (Single Page Application), тобто односторінкових додатків. В цьому плані Angular є спадкоємцем іншого фреймворка AngularJS. У той же час Angular це не нова версія AngularJS, а принципово новий фреймворк.

Angular надає таку функціональність, як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі.

Однією з ключових особливостей Angular є те, що він використовує в якості мови програмування TypeScript. Тому перед початком роботи рекомендується ознайомитися з основами цієї мови, про які можна прочитати тут.

Але ми не обмежені мовою TypeScript. При бажанні можемо писати програми на Angular за допомогою таких мов як Dart або JavaScript. Однак TypeScript все таки є основною мовою для Angular [10].

## 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Розробка блок-схеми, яка підлягає програмуванню

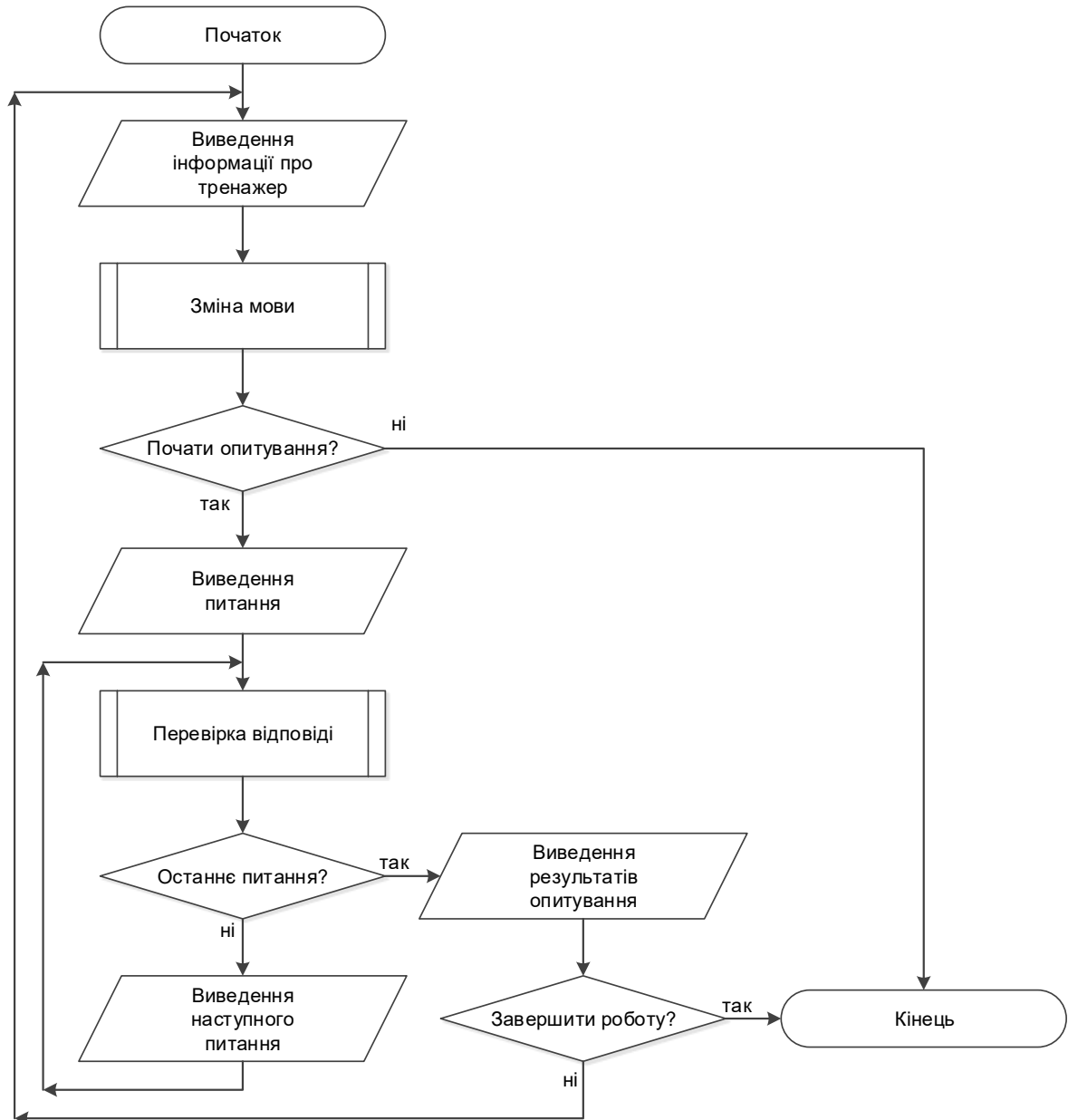


Рисунок 4.1 – Блок-схема роботи тренажеру

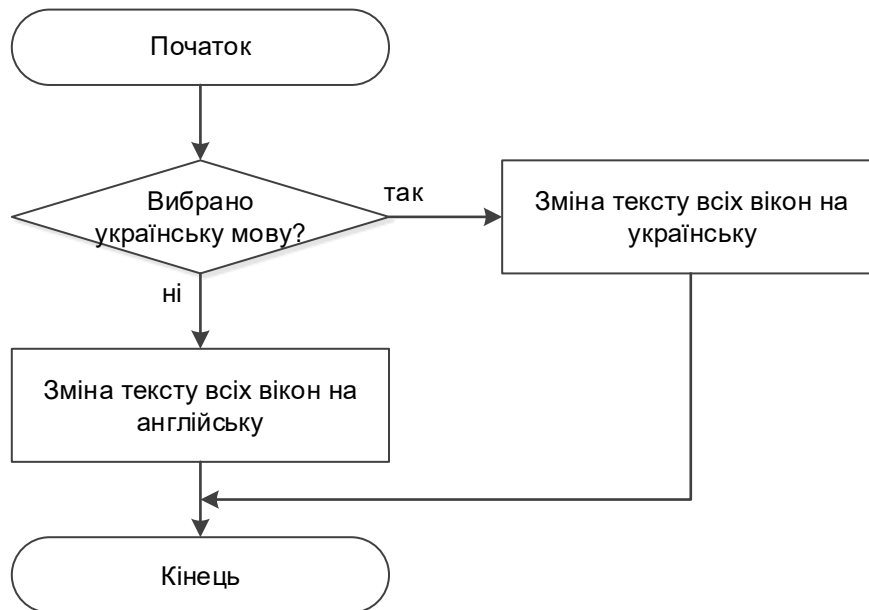


Рисунок 4.2 –Блок-схема зміни мови

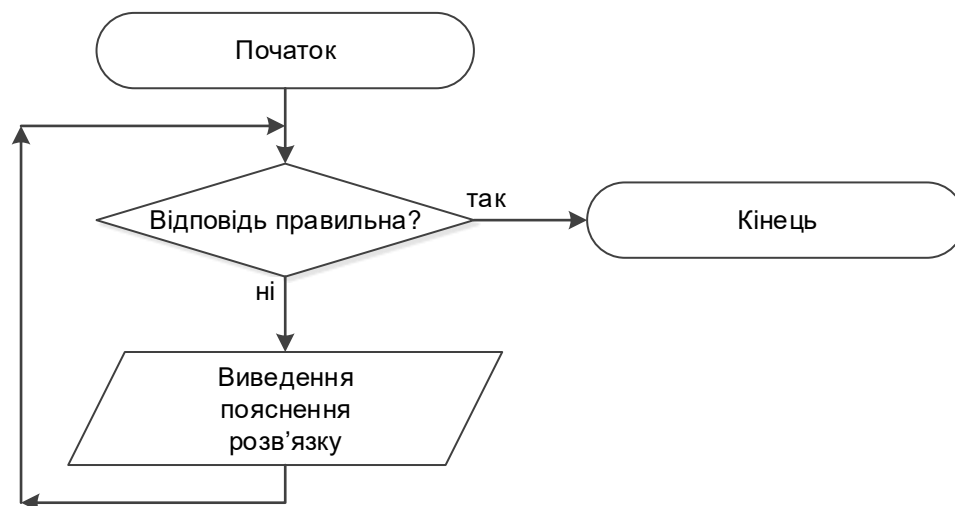


Рисунок 4.3 – Блок-схема перевірки відповіді

## 4.2. Опис процесу програмної реалізації

Навчальний тренажер на тему “Формальні граматики” розміщено на платформі Netlify за посиланням <https://puet-quiz.netlify.app/>. Netlify - це універсальна платформа для автоматизації сучасних веб-проектів.

Для створення тренажеру було обрано мову програмування Typescript, яка дозволяє робити статичну перевірку коду, з використанням фреймворку Angular. Фреймворк дозволяє створювати односторінкові

додатки для перегляду у браузері. Цей інструмент має також свою екосистему, яка дозволяє створювати динамічні шаблони для відображення контенту, стилізувати їх, використовуючи препроцесор SCSS для мови стилізації CSS.

Шаблон сторінки із запитанням:

```
<p>{{ questions[questionId].question }}</p>
<div [ngSwitch]="questions[questionId].type">
  <div *ngSwitchCase="'corresponding'" [formGroup]="answersMatch">
    <div *ngFor="let option of questions[questionId].options; let i = index;
trackBy: trackByFn"
      class="match">
      <div class="match-option">{{ option }}</div>
      <div class="match-input">
        <input matInput
          FormControlName="{{ i+1 }}">
      </div>
```

Виведення повідомлення помилки:

```
<div class="question-error">
  <div *ngIf="isError" class="question-error__content">
    {{ questions[questionId].error_phrase }}
  </div>
</div>
</div>
<br/>
```

Відображення кнопок навігації:

```
<div class="question-actions">
  <button mat-stroked-button
    color="primary"
```

```
(click)="submit()"
```

```
[disabled]="isAnswered">
```

Відповісти

```
</button>
```

```
<button mat-stroked-button
```

```
color="primary"
```

```
(click)="next()"
```

```
[disabled]="!isAnswered">
```

Далі

```
</button>
```

```
</div>
```

```
</div>
```

Виведення результату:

```
<div *ngIf="questionId > questions.length - 1" class="quiz-result">
```

```
<h2>Результат</h2>
```

```
<br/>
```

```
<div>
```

```
<span>Вітаємо! Ви пройшли всі тести з теми «Формальні  
граматики», Ваша оцінка: </span>
```

```
<span>{{getAssessment()}}</span>
```

```
</div>
```

```
<div>Ви дали {{answersCounter}}/{{questions.length}} правильних  
відповідей</div>
```

```
</div>
```

### 4.3. Опис роботи тренажеру

Тренажер складається з трьох типів сторінок. Перша сторінка пропонує користувачу почати пройти опитування з обраної теми (рис. 4.4):

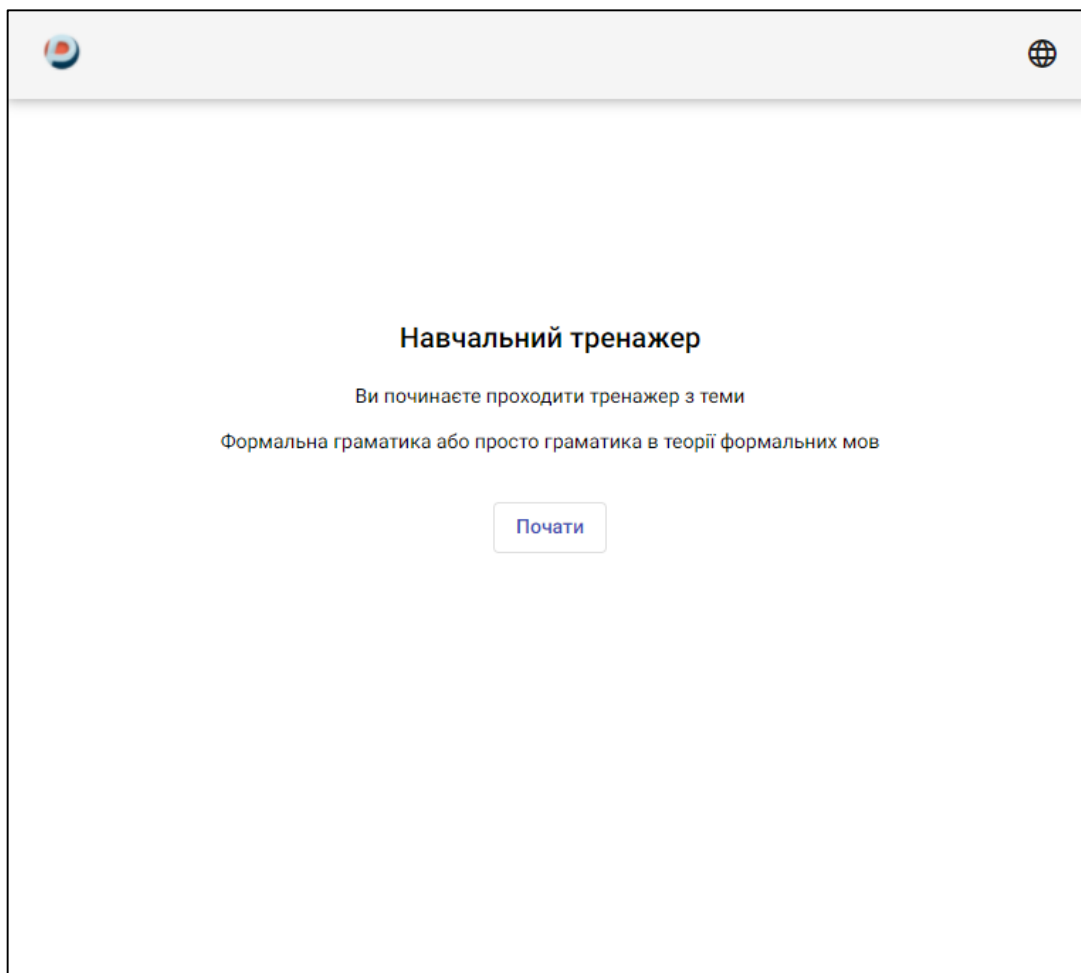



Рисунок 4.4 – Перша сторінка

Зміна мови програми здійснюється натисненням на значок  (рис. 4.5).

По кліку на кнопку “Почати”, користувач переходить до другого типу сторінок з відображенням питанням, яке може бути трьох видів:

- з однією можливою відповіддю;
- з багатьма можливими відповідями;
- із опціями, щоб встановити відповідності.

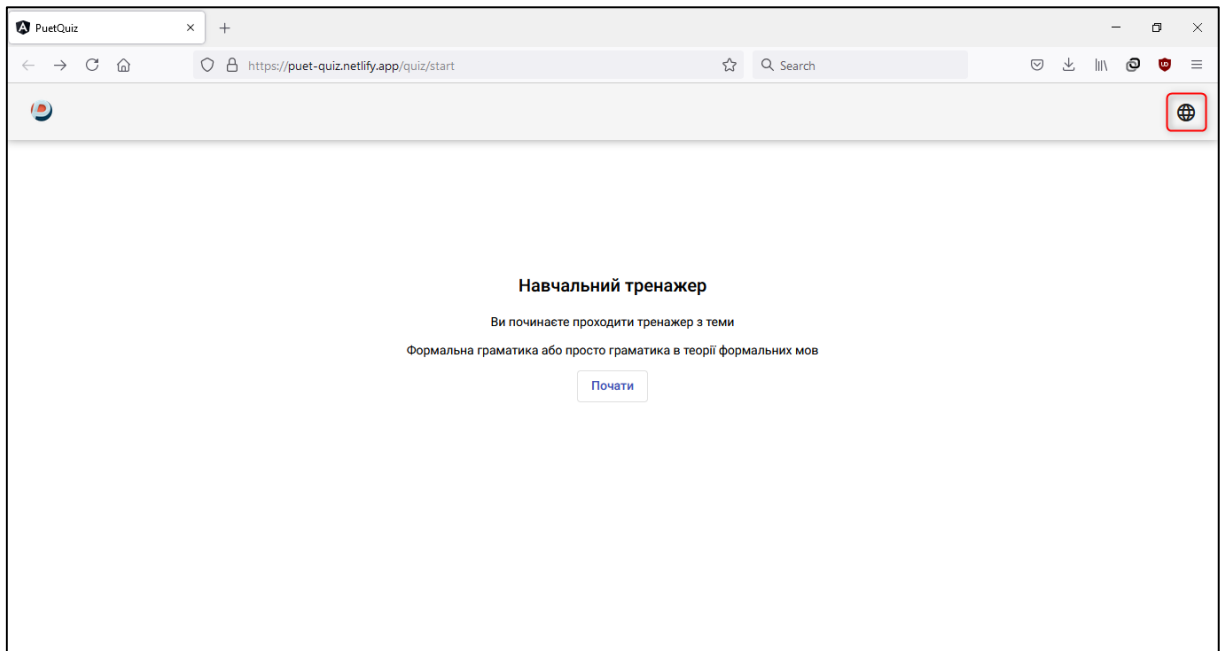


Рисунок 4.5 – Зміна мови

Приклад сторінки із питаннями (рис. 4.6):

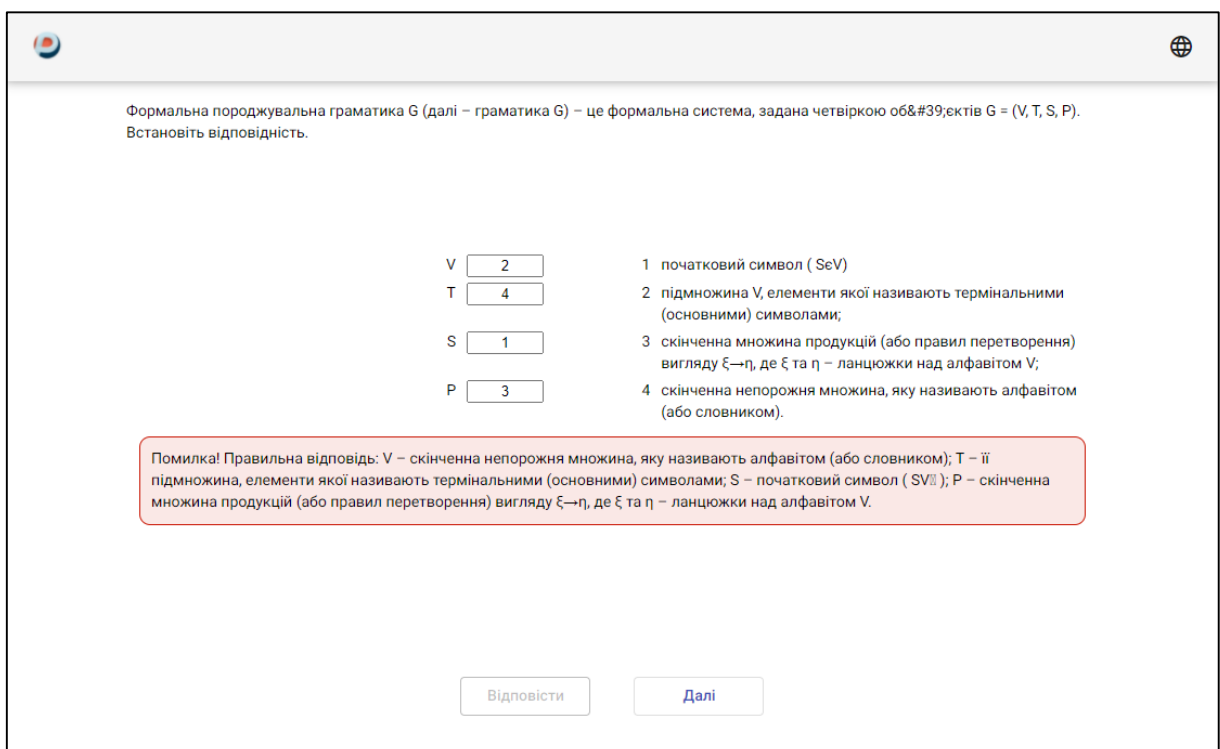


Рисунок 4.6 – Сторінка із питаннями

Користувачу пропонується питання і можливі відповіді до нього (рис. 4.7).

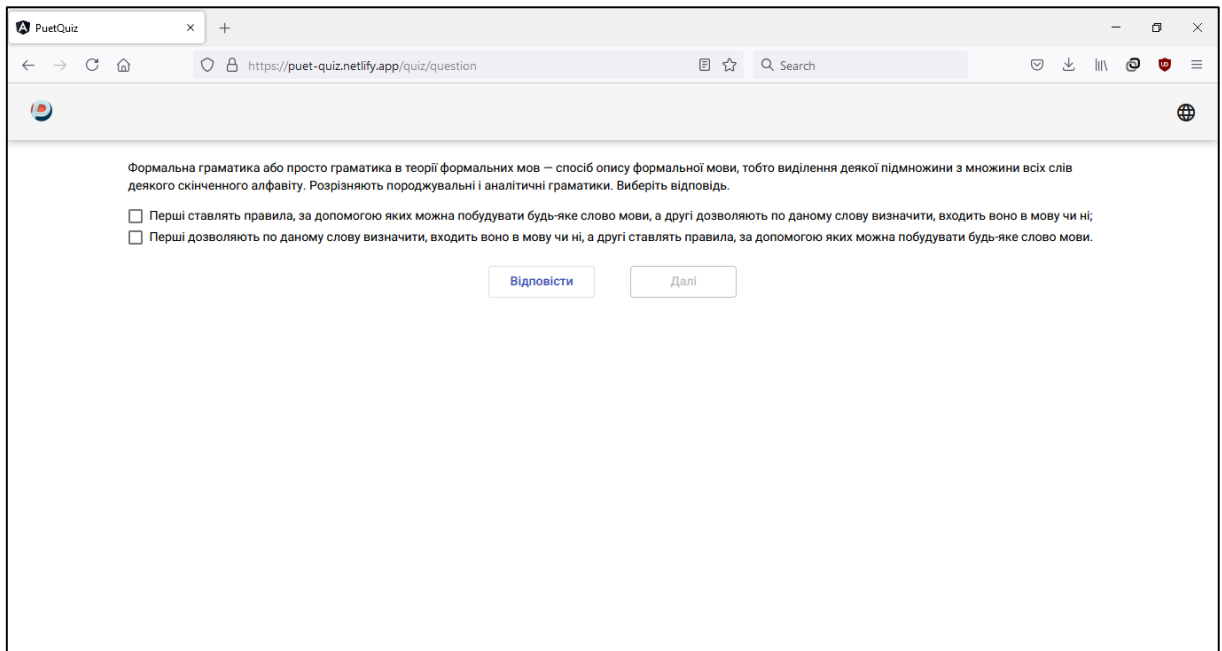


Рисунок 4.7 – Перше питання

Користувач має відмітити відповіді, які він вважає вірними, і натиснути кнопку «Відповісти» (рис. 4.8).

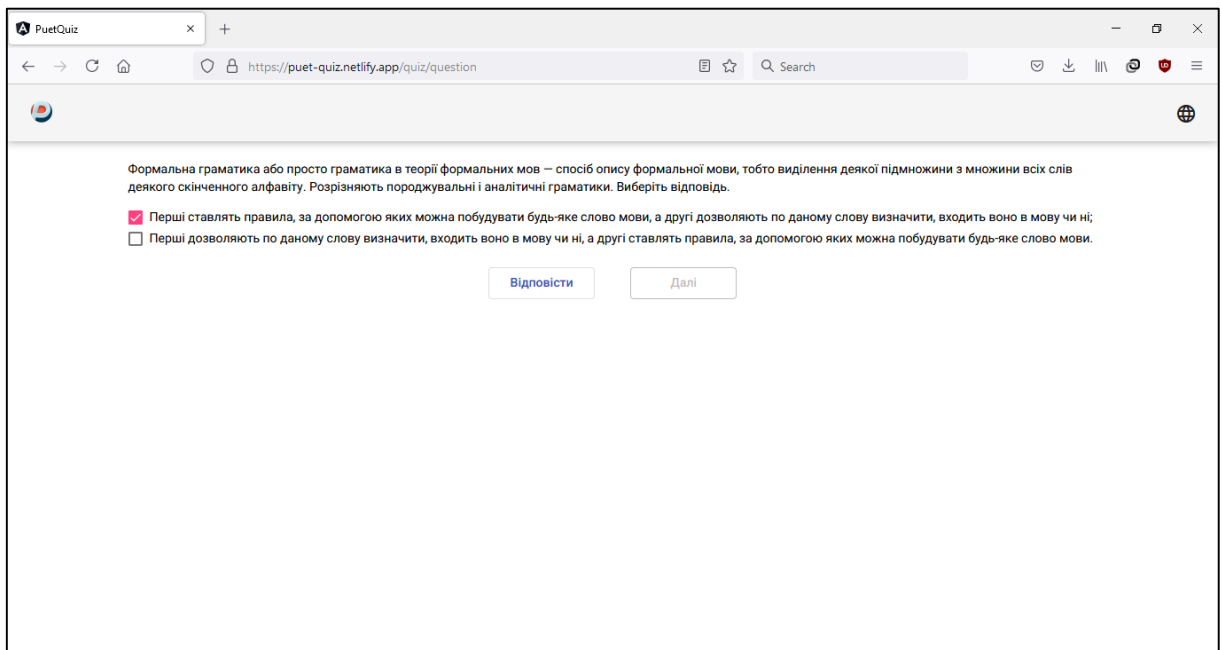


Рисунок 4.8 – Відповідь відмічено



Програма аналізує відповідь. Якщо відповідь неправильна, тренажер виводить текст помилки і приводить правильну відповідь, щоб користувач мав змогу запам'ятати правильну відповідь (рис. 4.9).

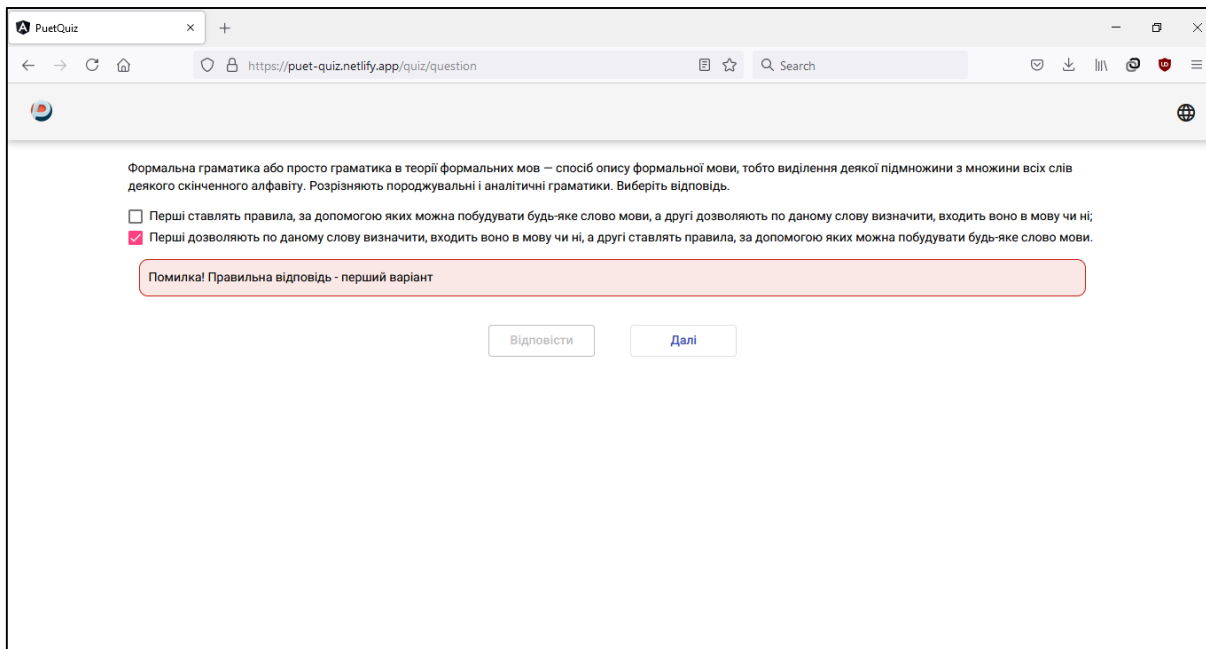


Рисунок 4.9 – Текст помилки

В такому випадку вона не зараховується, і користувач переходить до наступного питання (рис. 4.10) за допомогою кнопки “Далі”.

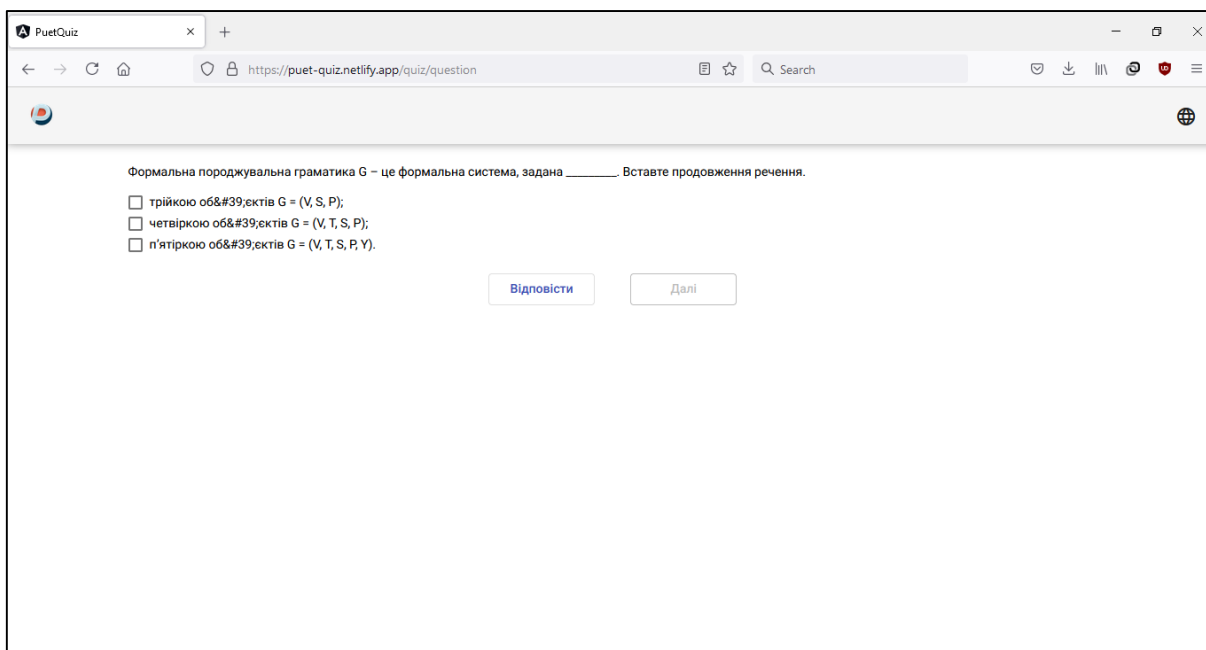


Рисунок 4.10 – Друге питання

По завершенню усіх питань, користувач переходить на сторінку із результатами пройденого опитування. (рис. 4.11).

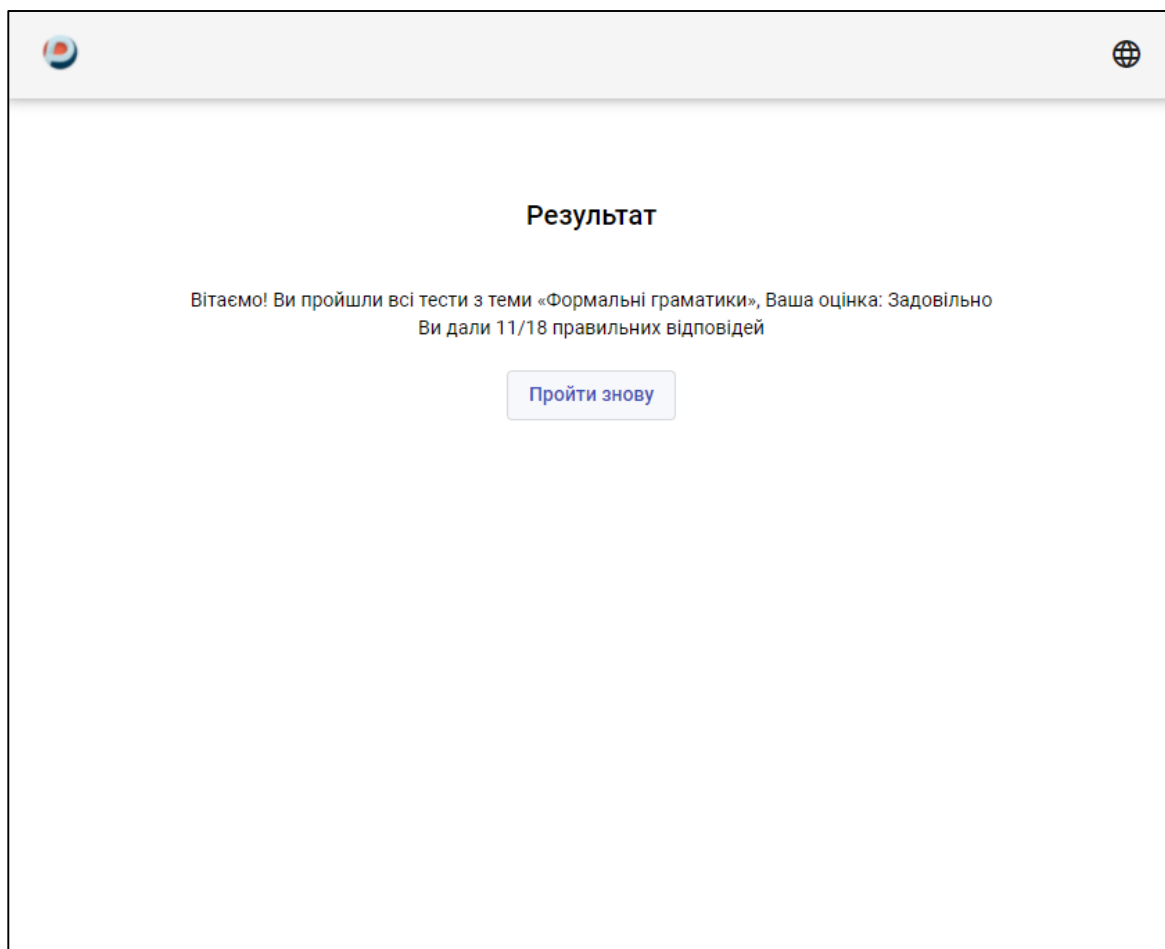


Рисунок 4.11 – Сторінка із результатами опитування

## ВИСНОВКИ

Розглянемо виконані завдання проекту:

- описано постановку задачі;
- описано інформаційний огляд;
- переглянуто теоретичний матеріал з теми та наведено основні поняття для використання в тренажері;
- розроблено алгоритм тренажеру;

В алгоритмі користувачу описано процес тестування. На кожному кроці виводиться умова, потім надається один із способів:

- вибір варіантів відповіді;
- встановлення відповідності;
- встановлення послідовності кроків.

При неправильній відповіді відображається помилка.

Завершивши всі тести виводиться відповідне повідомлення: «Вітаємо! Ви пройшли всі тести з теми «Формальні граматики», Ваша оцінка:», вказується один із варіантів в залежності від кількості допущених помилок:

- Відмінно;
- Дуже добре;
- Добре;
- Задовільно;
- Незадовільно.

Загалом, ДО в Україні не відповідає вимогам, що ставляться до інформаційного суспільства і не забезпечує повноцінного входження України в міжнародний освітній простір. Щоб система дистанційного навчання зайняла гідне місце в системі освіти України, потрібно, передовсім, створити глобальну комп'ютерну мережу освіти й науки, оскільки саме комп'ютер дає змогу отримувати навчальний матеріал, є водночас і бібліотекою, і центром довідкової інформації, і комунікативним

центром, що робить його одним з учасників реалізації програми безперервної освіти в Україні.

Отже, дистанційне навчання є досить новим явищем в Україні, та воно швидко розповсюджується серед вітчизняних ВНЗ. Серед великої кількості переваг у нашій країні воно зіткнулося з низкою проблем, адже новітні технології не є досконалими та рівень викладачів не забезпечують швидке освоєння та користування цим ресурсом. Дистанційна освіта в Україні повинна переймати досвід європейських країн для швидкого його розвитку та реформування.

Оскільки тренажер буде використовуватися іноземними студентами в рамках вивчення дистанційного курсу «Теорія програмування», то розроблено можливість зміни мови під час виконання.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології» галузь знань - 12 «Інформаційні технології» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2011. – 71 с.
2. Андреев А.А. Дистанционное обучение: сущность, технология, организация / А.А. Андреев, В.И. Солдаткин. – М.: Изд-во МЭСИ, 2000. – 350 с.
3. Вільне програмне забезпечення в освіті [Електронний ресурс]. – Режим доступу: <http://edufoss.blogspot.com/search/label>
4. Кравчина О.Є. Основні напрямки використання вільного програмного забезпечення [Електронний ресурс] / О.Є. Кравчина // Інформаційні технології і засоби навчання. – 2010. – №6(20). – Режим доступу: <http://www.ime.edu-ua.net/em.html>
5. Смирнова-Трибульська Є.М. Дистанційне навчання з використанням системи MOODLE : навч.-метод. посіб. / Є.М. Смирнова-Трибульська. – Херсон: Видавництво Айлант, 2007. – 465 с.
6. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика» – Режим доступу: <http://tprogr.ho.ua>.
7. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.– Суми: Вид-во СумДУ, 2009. – 181 с.
8. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://ttp.unicyb.kiev.ua/doc/TOP.pdf>.

9. TypeScript [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: <https://uk.wikipedia.org/wiki/TypeScript>.
10. Введение в Angular. Что такое Angular. Первый проект [Електронний ресурс] – Режим доступу: <https://metanit.com/web/angular2/1.1.php>.
11. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

**ДОДАТОК А**